

Article

CNN-Based Suppression of False Contour and Color Distortion in Bit-Depth Enhancement [†]

Changmeng Peng ¹, Luting Cai ², Xiaoyang Huang ¹, Zhizhong Fu ^{1,*}, Jin Xu ¹ and Xiaofeng Li ¹

¹ School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), 2006 Xiyuan Avenue, Chengdu 611731, China; pengcm@std.uestc.edu.cn (C.P.); hxy@std.uestc.edu.cn (X.H.); jinxu@uestc.edu.cn (J.X.); xfli@uestc.edu.cn (X.L.)

² ZTE Corporation, Chongqing 401121, China; clt19951229@gmail.com

* Correspondence: fuzz@uestc.edu.cn

[†] This paper is an extension version of the conference paper: C. Peng, L. Cai, Z. Fu and X. Li, “CNN-based bit-depth enhancement by the suppression of false contour and color distortion”, 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Lanzhou, China, 2019; pp. 1145–1151, doi:10.1109/APSIPAASC47483.2019.9023201.

Abstract: It is a challenge to transmit and store the massive visual data generated in the Visual Internet of Things (VIoT), so the compression of the visual data is of great significance to VIoT. Compressing bit-depth of images is very cost-effective to reduce the large volume of visual data. However, compressing the bit-depth will introduce false contour, and color distortion would occur in the reconstructed image. False contour and color distortion suppression become critical issues of the bit-depth enhancement in VIoT. To solve these problems, a *Bit-depth Enhancement method with AUTO-encoder-like structure* (BE-AUTO) is proposed in this paper. Based on the convolution-combined-with-deconvolution codec and global skip of BE-AUTO, this method can effectively suppress false contour and color distortion, thus achieving the state-of-the-art objective metric and visual quality in the reconstructed images, making it more suitable for bit-depth enhancement in VIoT.

Keywords: bit-depth enhancement; Visual Internet of Things; convolution neural network; false contour; color distortion



Citation: Peng, C.; Cai, L.; Huang, X.; Fu, Z.; Xu, J.; Li, X. CNN-Based Suppression of False Contour and Color Distortion in Bit-Depth Enhancement. *Sensors* **2021**, *21*, 416. <https://doi.org/10.3390/s21020416>

Received: 21 December 2020

Accepted: 4 January 2021

Published: 8 January 2021

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Visual sensors can provide richer and more intuitive information compared to other types of sensors. They are critical *Perception Front Ends* (PFEs) of VIoT, and have been used in many scenarios, such as security surveillance, person identification, image retrieval, and telemedicine [1,2]. However, compared with other types of signals, such as temperature sensor signals, visual signals have a huge amount of data. The surveillance system consisting of 6 million cameras has an hourly data volume of about 7.5 petabytes [3]. It is a great challenge to transmit and store these visual data.

Usually, each PFE in VIoT can only supply the compact memory space, limited energy and computing resources. Therefore, the visual signals perceived by PFE must be sent to the cloud platform with massive storage space and rich computing resources [1]. Even with the help of high-efficient compression technology, such as H.26x, in VIoT application environment, compression of massive VIoT data will occupy a lot of bandwidth and consume a lot of energy, which is a heavy burden for the green environmental system. Therefore, based on the existing video data encoding and decoding framework, it is very necessary to do proper preprocessing before the visual data is compressed [4].

Existing preprocessing methods have high complexity and are not suitable for the limited resource application environment of VIoT. The advantage of compressive sensing (CS) is that it can perform data sampling and compression at the same time, which makes

it widely investigated in IoT scenarios [5–9]. However, resources required to calculate, store the measurement matrix and compression process are huge burdens for VIoT. With the popularity of deep learning, *Convolution Neural Network* (CNN) feature maps are used to encode images [4]. However, this kind of CNN-based preprocessing method requires the deployment of CNN on the sensor, which greatly increases the complexity and cost of the sensor, which is not suitable for VIoT.

In contrast to CS and CNN-based preprocessing, reducing the bit-depth of images can achieve a significant reduction in the amount of visual data with minimal computational overhead. For example, compressing an image with a bit-depth of 16 bits to 4 bits can achieve 75% data compression, and the extra calculation required at the encoder side is only to discard the extra least significant bits. At the decoder side, *Bit-Depth Enhancement* (BDE) technology is used to restore visual signals which have almost the same visual perception quality as the visual perception front end. The asymmetric computing model based on this simple encoding and complex decoding is very suitable for the characteristics of the IoT system which is built from resource-limited PFE and resource-rich computing platforms. It should be noted that compression bit-depth may introduce false contours in the low-bit image. Therefore, false contours must be eliminated as accurately as possible without introducing other artifacts during the reconstruction process.

There are already several BDE algorithms, including context-independent methods, such as *Multiply by Ideal Gain* (MIG) [10] and *Minimum Risk-based Classification* (MRC) [11]. This kind of methods do not make use of context information, so they are not good at suppressing false contours. The context-dependent methods represented by *Contour Region Reconstruction* (CRR) [12] and *Intensity Potential for Image De-quantization* (IPAD) [13] improves the false contour suppression effect by using the information of surrounding pixels, but some non-contour edges are blurred. Methods based on deep learning, such as *Bit-depth Enhancement via Residual Transposed Convolutional Neural Network* (BE-RTCNN) [14,15] and *Deep Reconstruction of Least Significant Bits* (BDEN) [16], use the powerful non-linear fitting capabilities of deep learning to push the performance of the BDE algorithm to a new height. However, more effective network structures, loss functions and training strategies are still worth exploring.

In this paper, we propose a deep convolution network-based BDE method BE-AUTO. It has an auto-encoder structure, where the encoder and decoder are implemented by 5 convolution and deconvolution layers, respectively. This implementation of convolution-combined-with-deconvolution is better than pure convolution or pure deconvolution. In order to achieve better data flow and mitigate the problem of gradient vanishing in deep networks, we add skip connections between the corresponding layers of the encoder and decoder. By using the visual loss proposed in [17] (hereinafter, we collectively call it *vgg_loss*), BE-AUTO can better suppress false contours than using Mean Square Loss. We find color distortion in some of the result images and that this distortion is caused by out-of-bounds values. To solve this distortion, we add a global skip from the input layer to the output in the network structure. The introduction of the global skip greatly reduces the estimation error, thereby effectively solving the out-of-bounds problem and color distortion. Thanks to BE-AUTO's convolution-combined-with-deconvolution auto-encoder structure, global skip and *vgg_loss*, our algorithm can better suppress false contours and color distortion compared with previous algorithms thus achieving the best objective metrics and subjective visual quality.

The main contributions of this article are as follows:

- An auto-encoder of convolution-combined-with-deconvolution structure is proposed for BDE, and it is superior to pure convolution and pure deconvolution. With the help of the global skip connection, the color distortion is well suppressed.
- The mechanism of color distortion is analyzed in detail. It has been experimentally proved that the value of the restored image can be confined into a reasonable range by applying a global skip connection to suppress the color distortion. Moreover, it has

been also proved by experimental results that global skip is effective under different network structures.

- A deep BDE algorithm BE-AUTO is proposed. It can effectively suppress false contours and color distortion under the constraint of *vgg_loss*, and obtain state-of-the-art experimental results both in objective and subjective performance.

The remainder of this paper is organized as follows. Section 2 gives related works and presents our motivation for this work. Section 3 describes our proposed bit-depth enhancement method. The experimental results are presented in Section 4, with ablation analysis in Section 4.3. Section 5 concludes this paper.

2. Related Work

BDE is designed to eliminate as many false contours as possible in low-bit-depth input images when reconstructing bit-depth [10,11,18–26]. Effective bit-depth enhancement algorithms are essential for energy-efficient VIoT based on bit-depth compression, but the existing BDE algorithms either cannot suppress false contours well or lose much detailed information, so they are not suitable for visual IoT scenarios.

The existing bit-depth enhancement algorithms can be divided into traditional methods and deep-learning-based methods. Traditional methods can be roughly divided into context-independent and context-dependent. Compared with the context-independent methods, the context-dependent methods can better suppress false contours, but have higher complexity. There are few existing BDE algorithms based on deep learning, but they have achieved better results than traditional methods.

Context-independent BDE methods only consider the information of the current pixel when extending the bit-depth. Among them, the easiest way is *Zero-Padding* (ZP). In the ZP algorithm, the *least significant bits* (LSBs) of the input image are directly filled with zeros to achieve the target bit-depth. MIG [10] algorithm multiplies the image value of the low-bit-depth image by an ideal gain: $(2^p - 1)/(2^q - 1)$ where p, q are the expected bit-depth and the bit-depth of the input image, respectively. *Bit Replication* (BR) [10] is a simple approximation of the MIG algorithm. It takes the q bits of the input image as a whole, periodically attaches it to the latest LSB until the bit-depth is not less than p , and then outputs the most significant p bits. MRC [11] models bit-depth extension as a minimum classification risk problem. MRC generates an error distribution function of all the possible estimation values and accepts the value that minimizes the associated risk. Context-independent algorithms have low algorithm complexity since they only consider the current pixel. However, due to the lack of use of context information, false contours cannot be effectively suppressed.

Context-dependent BDE methods take into account the context information of the current pixel to effectively suppress false contours. The algorithm *maximum a posteriori estimation of AC signal* (ACDC) [27,28] divides image signal into AC and DC components by constructing graph-Laplace. AC component is estimated based on the *maximum posterior probability* (MAP), which is then used to help the estimation of DC. Because the image's graph-Laplace takes into account the context information of the pixel, ACDC suppresses false contours better than the context-independent. However, due to complexity considerations, ACDC must divide the image into small patches, which makes it can only use the context information of the current image patch, so the suppression effect is not good. CRR [12] states that if the value of at least one pixel in the four neighborhoods of a pixel is less/greater than that pixel, the pixel is called the upper/lower boundary pixel. The set of adjacent upper/lower boundary pixels is called the upper/lower boundary. CRR finds out all upper/lower boundaries and then estimates the pixel value of the current location based on *step_ratio* defined by the distances from the current pixel to its closest upper boundary and lower boundary. CRR algorithm could suppress false contours better than previous algorithms. However, in the local maximum/minimum regions, the upper or lower boundary does not exist. Therefore, the false contour suppression effect of these regions is bad. To solve this issue, *Content-Adaptive Image Bit-depth Expansion* (CA) [29] in-

roduces virtual skeletons in these areas to build virtual boundaries. Moreover, CA defines a different *step_ratio* according to different pixel categories (such as local maximum area pixels, absolute dark pixels), and implements content-adaptive at some level. Regardless of CRR or CA, for a pixel, only the closest upper boundary and lower boundary are active. However, all the boundaries of the flat area where the pixel is located should work for the estimation of the pixel value. Based on this assumption, IPAD [13] makes all the boundaries of the flat region act on the estimation of the current pixel by using the additivity principle of potential, and thus achieving better results. CRR, CA, IPAD improve the suppression of false contours by using context information. However, by converting the estimation of the pixel value of the 2-D image into a 1-D interpolation, the context information of the 2-D image cannot be fully exploited and used. Moreover, CRR and CA update the distance by flooding, making them extremely time-consuming.

Compared with the wide application of deep learning in other image processing fields (such as image super-resolution [30–34]), there are few studies on BDE algorithms based on deep learning. As far as we know, there are only three deep image BDE algorithms: BE-RTCNN [14,15], BDEN [16] and *Concatenating All Level Features* (BE-CALF) [35]. Among them, BE-RTCNN used a full deconvolution residual network and visual loss (loss function based on the visual-geometry-group network (VGG) [36]) to achieve the suppression of false contours. BDEN was a two-column residual network that was used to process flat and non-flat areas in an image, respectively. Because the magnitude of the residuals to be learned was very small, BDEN scaled up the outputs of the two branches separately. Compared with traditional methods, BE-RTCNN and BDEN had achieved better objective indicators and subjective quality. However, the presence of false contours and color distortion could still be seen in the resulting image.

Our method is a little similar to BE-CALF, both using auto-encoder-like network structure and perceptual loss. The difference is that BE-CALF used dense skip connections, but in our method, more efficient symmetric skip connections are used. Another obvious difference is that BE-CALF did not pay attention to the problem of color distortion.

This article is an extension of our previous conference paper [37]. The contents of the extension mainly include the following aspects:

- We make an in-depth analysis of the causes of color distortion, and based on numerical interval estimation and estimation error analysis, theoretical support is provided for the final solution;
- Experiments are made to verify that the global skip connection has similar color distortion suppression effect for different network structures;
- The test set is extended. We add 3 test sets to make a sufficient comparison of different scale BDE in natural and synthetic images between our algorithm and the related;
- The latest related algorithms and 4 visual comparison figures are added to fully reflect the merits of our algorithm.

3. Materials and Methods

3.1. Problem Model

3.1.1. Quantization and Bit-Depth Variation

The quantizer is an important part of the conversion of image signals from analog to digital. It quantifies the continuous intensity of the light into discrete values, which are then represented by a finite number of bits. The number of bits used per pixel is called the bit-depth of the image, and a common bit-depth is 8 bits. The deeper the bit-depth is, the more realistic the visual quality can be obtained, and of course, the more transmission and storage resources are required.

Quantization methods can be divided into linear quantization and non-linear quantization. Linear quantization is the simplest and most commonly used quantization method. To reduce the number of bits per pixel of an image, a high-bit image needs to be converted into a low-bit image. The conversion process is essentially a re-quantization of the image data. When a high-bit (p) image is quantized to a low-bit (q) image with the use of

linear quantization, a low-bit image can be obtained by discarding g LSBs of the high-bit image, where $(g = p - q)$ represents the bit-depth difference. BDE is the reverse of this quantization process. Therefore, BDE is also called image reverse quantization [13].

3.1.2. Key Problems to Be Solved

Under the condition of limited bit-depth, if the values of adjacent pixels in the smooth area of the image fall into different quantization intervals, after quantization, an incorrect contour will appear between adjacent pixels. The smaller the quantized bit-depth, the more obvious the false contour. As shown in Figure 1, the lower half of the figure is a high-bit image with a bit-depth of 16 bits, and the upper half is the 4-bit version. As we can see, obvious false contours appear in the low bit-depth image. False contours will severely degrade the visual quality of the image. Therefore, BDE should not only extend the bit-depth but also suppresses false contours, so that the restored image has better visual quality and is as similar to the original high-bit image as possible.



Figure 1. False contour example [37]. The upper part is the reconstructed 16-bit image from 4-bit via ZP and the lower part is the 16-bit ground-truth.

Apart from false contour artifacts, color distortion (as shown by Figure 8c) is another problem to be solved. The direct cause of color distortion is that the estimated values of some components in RGB deviate significantly from the true values. In regression-based BDE methods, such as CNN-based ones, this deviation is more likely to occur, because the optimization target is to minimize the average estimation error.

3.1.3. Mathematical Modeling

As shown in Equation (1), the core purpose of BDE is to find a non-linear mapping function f that converts a low-bit input image into a high-bit one. Moreover, during the conversion process, the similarity between the output image and the original high-bit image is maximized with the help of image prior knowledge. In Equation (1), $I^{H/LBD}$ is HBD/LBD image, s is similarity measure function, and pr represents image prior. Commonly used image priors include sparsity prior [25], smoothness prior such as ACDC [28] and image content context prior such as CRR [12], CA [29], IPAD [13].

$$\arg \max_f s(I^{HBD}, f(I^{LBD})) \quad s.t. \quad pr(I^{LBD}) \quad (1)$$

Deep learning is a data-driven approach. With the help of the loss function and gradient descent, the deep learning network gradually learns the prior information described by the training sample set, and stores it in the form of parameters of the deep network. Therefore, when using the deep network model, the BDE process described in Equation (1) will be rewritten as follows:

$$\arg \min_{\theta} loss(I^{HBD}, CNN_{\theta}(I^{LBD})) \quad (2)$$

where θ represents the network parameters. It should be noted that the prior term in Equation (1) is embodied in the specific form of the loss function in Equation (2). At the same time, the convolutional neural network itself is powerful prior [38].

3.2. Loss Function

Loss function is an important part of the neural network. It directly affects the search direction of the network, which decides the network convergence speed and the final performance [39]. Image super-resolution is a field similar to BDE. In image super-resolution, the commonly used loss function is the Mean Square Error loss (*mse_loss*) [30,32], which is the mean square error of the output image and the label image. *mse_loss* is widely used in super-resolution, and it has achieved good results. However, false contours cannot be suppressed well [14] in BDE with *mse_loss* (defined by Equation (3), where W, H, I, \hat{I} are width, height, ground-truth and reconstruction result of an image respectively). In SRGAN [17], the loss function is defined on the feature maps of the output image and the label image in a pre-trained VGG network [36], which allows the resulting image to have more natural texture details. Inspired by this, we define a loss function *vgg_loss* (defined by Equation (4), where G, I, J are the pre-trained VGG, index of feature map and channel of VGG respectively) on the VGG feature map as the optimization goal of BE-AUTO. It should be noted that *vgg_loss* is also applied in BE-RTCNN, where it is called visual loss.

$$mse_loss = \frac{1}{WH} \sum_{x=1}^W \sum_{y=1}^H (I_{x,y} - \hat{I}_{x,y})^2 \quad (3)$$

$$vgg_loss = \sum_{i=1}^I \sum_{j=1}^{J_i} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} \frac{1}{I_j W_{i,j} H_{i,j}} (G(I)_{x,y}^{i,j} - G(\hat{I})_{x,y}^{i,j})^2 \quad (4)$$

3.3. BDE Model

To obtain results of excellent subjective and objective quality, the deep BDE algorithm must suppress the false contour and overcome the color distortion at the same time.

The proposed scheme is presented in Figure 2. As shown in Figure 2, BE-AUTO is a convolution neural network-based bit-depth enhancement method with an auto-encoder-like structure. It takes RGB images as input and output RGB ones. It consists of an encoding module (left part) and a decoding module (right part). The encoding module is implemented by 5 (can be increased or decreased as needed) convolution layers. The size of convolution kernels (k^*) are all 3×3 , and from the first layer to the fifth layer, the number of convolution kernels (m^*) is doubled every layer from 16 to 256 to extract richer features. The decoding module is implemented by 5 deconvolution layers [40], and the size of the deconvolution kernel is also 3×3 . The number of convolution kernels is 128, 64, 32, 16, 3 from the fifth layer of the deconvolution to the first. Details will lose if the sizes of feature maps decrease, so we set stride (s^*) as 1. To alleviate the disappearance of the gradient and facilitate the training of the deep network, we add a *Batch Normalization* (BN) layer [41] behind each deconvolution layer and add a skip connection [42] between the corresponding layers of the codec module. The activation functions used in BE-AUTO are ReLU [43]. The reason the convolution and deconvolution are used to implement the codec module is that the convolution and deconvolution functions fit well with the codec, which will be analyzed and proved in Section 4.3.5. Based on the analysis of color distortion, we short connect the input to the output of the decoding module, and element-wisely add them up to get the final output. To distinguish it from the connection mentioned earlier, we call this connection “global skip”. Global skip will play a key role in suppression color distortion as shown in Section 4.3.4, because it can significantly reduce estimation errors (see Section 4.3.1 for more details).

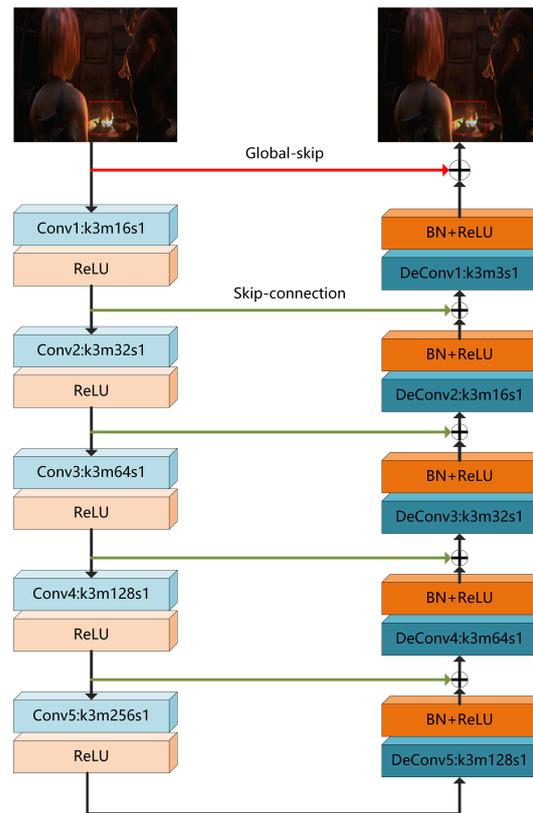


Figure 2. The network structure of the proposed algorithm BE-AUTO [37].

4. Experiments and Discussion

4.1. Experiment Settings

4.1.1. Data Sets

The test sets include sintel-8, fiveK-40, BDE-set, Kodak dataset [44]. Sintel-8 contains eight 16-bit images randomly selected from sintel [45] for testing large-scale artificial image BDE from 4-bit to 16-bit. The fiveK-40 consists of forty 16-bit natural images randomly selected from fiveK [46] for testing 4-bit to 16-bit enhancement of natural images. BDE-set and Kodak datasets contain eight and twenty-four 8-bit images respectively for testing small-scale BDE from 6-bit to 8-bit. BDE-set is a commonly used dataset in traditional BDE (like [11]). Since its quantity is small, BDE-set is supplemented with Kodak dataset.

Two training sets sintel-2000 and DIV2K [47] are used to train the network for 4-bit to 16-bit and 6-bit to 8-bit BDE, respectively. The sintel-2000 has 2000 frames selected from [45] and each frame has a resolution of 436×1024 and a bit-depth of 16. Sintel-2000 has no intersection with sintel-8. The DIV2K contains eight hundred high-quality 8-bit natural images with a resolution of 1404×2040 .

4.1.2. Algorithms to be Compared

The proposed algorithm is compared with the representative BDE methods, including: zero-padding (ZP), MIG [10], BR [10], MRC [11], ACDC [27,28], CRR [12], CA [29], IPAD [13], BE-RTCNN [14,15] and BDEN [16]. Of these methods, only BE-RTCNN and BDEN are deep learning-based methods, and the rest are traditional methods. It should be noted that the experimental data of BDEN and IPAD are the results of running the code provided by the papers' authors.

4.1.3. Training Details

The network training process is performed on an NVIDIA 1050Ti. A 96×96 patch is randomly cropped from each image. These patches form batches of size 8, and are fed into the network for training. Epoch is set to 120 for a good tradeoff between training time

and performance. The learning rate is set to 1×10^{-4} . As for optimizer, we choose Adam with $\beta_1 = 0.9$. Each 16/8-bit image is linearly quantified into a 4/6-bit image. Then, the 4/6-bit image is restored back to a 16/8-bit image through the Zero-Padding method, then fed into the network for training.

4.2. Results and Discussion

4.2.1. Objective Performance Evaluation

Table 1 lists the PSNRs (computed on RGB color space) for all methods in comparison on the test sets sintel-8, fiveK-40, BDE-set and Kodak. The best result is highlighted in bold and the second best is underlined. A comparison of the PSNR curves on test datasets is shown in Figure 3.

Table 1. Comparison with state-of-the-art on PSNR.

Dataset	ZP	MIG	BR [10]	MRC [11]	ACDC [28]	CRR [12]	CA [29]	IPAD [13]	BE-RTCNN [15]	BDEN [16]	Ours
sintel-8	29.7141	31.6352	32.1936	33.0923	33.3474	31.3352	34.9920	34.4867	<u>35.2994</u>	-	36.2270
fiveK-40	28.8342	31.3927	31.9534	33.6832	34.8084	33.8474	35.2481	35.5903	34.3688	-	36.0413
BDE-set	42.7506	45.4445	44.3908	<u>47.0334</u>	47.0099	45.0150	45.9547	45.5052	-	-	47.1877
Kodak	42.5077	45.5028	45.1540	<u>47.1177</u>	47.1001	45.1564	46.0462	45.5074	-	46.4550	47.2576

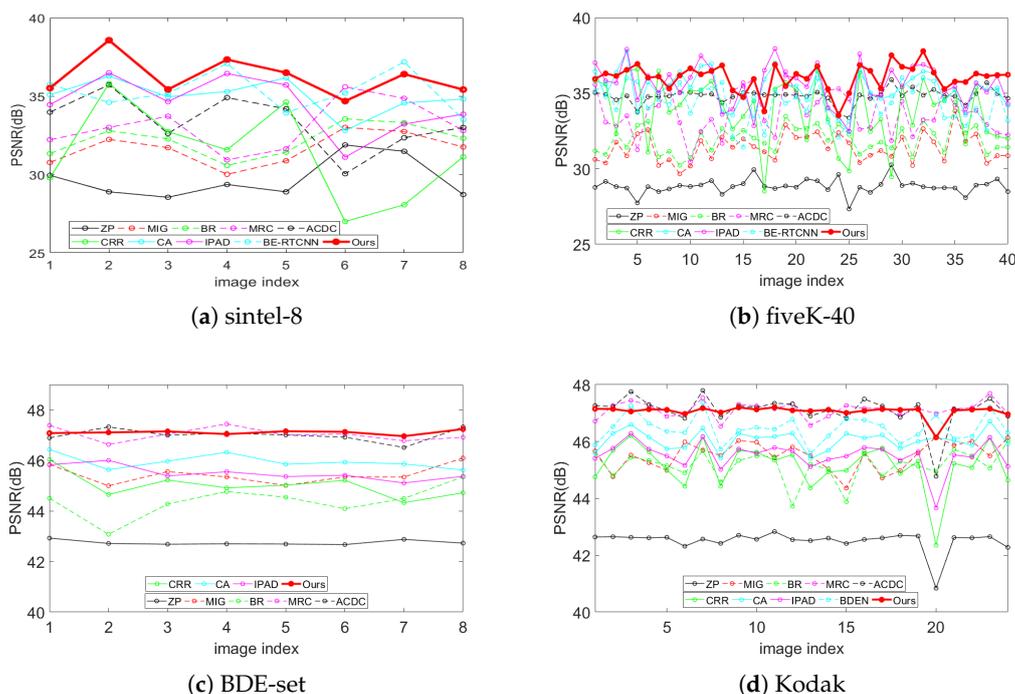


Figure 3. PSNR curves comparison with state-of-the-art on four test datasets.

It can be seen from Table 1 and Figure 3 that our method achieves superiority to other algorithms in large-scale BDE scenes from 4-bit to 16-bit for both natural images (fiveK-40) and artificial images (sintel-8). In small-scale scenarios from 6-bit to 8-bit, MRC and ACDC show significant advantages over other traditional methods, especially on the Kodak test set. This is most likely because that in this scenario, there are only four possible values for the newly added two bits, which greatly reduces the risk of classification error of MRC and the error probability of the AC component of ACDC. This is just the reason these two algorithms do not perform well in large-scale scenarios. For small-scale scenarios, our algorithm has achieved the best results on both BDE-set and Kodak, with a gain of 0.15 dB compared with the second-best algorithm. In summary, our algorithm achieves better objective results than these comparison algorithms for all scales and image types. This is due to the following three aspects:

- (1) The use of convolution for encoding and deconvolution for decoding can better extract features and perform better reconstruction.
- (2) The addition of the global skip connection allows the network output to remain well within the desired range, thereby avoiding color distortion and not interfering with the false contour suppression.
- (3) *vgg_loss* has an incomparable inhibitory effect on false contours relative to *mse_loss*, which is fully discussed in BE-RTCNN [15].

4.2.2. Subjective Performance Evaluation

Subjective performance comparisons with state-of-the-art are depicted in Figures 4–7. Where Figures 4 and 5 show the comparison of artificial image BDE on a large scale (from 4-bit to 16-bit). Large-scale natural image BDE comparison from 4-bit to 16-bit is depicted in Figure 6. For 6-bit to 8-bit small-scale BDE (Figure 7), the difference between the 6-bit image and the 8-bit ground-truth is already hard to distinguish (as depicted by Figure 7a,b) by human eyes, so we use pixel value curves to show the reconstruction's similarity with the ground-truth [16].

It can be seen from Figure 4 that the results of the traditional methods have different degrees of false contours. Although CRR suppresses false contours well, the results are over-smoothed (such as the hilltop). Compared with the traditional method, under BE-RTCNN, false contours can be suppressed well, but significant color distortion occurs. That is because of the lack of global skip connection to effectively limit the range of values, resulting in numerical flipping.

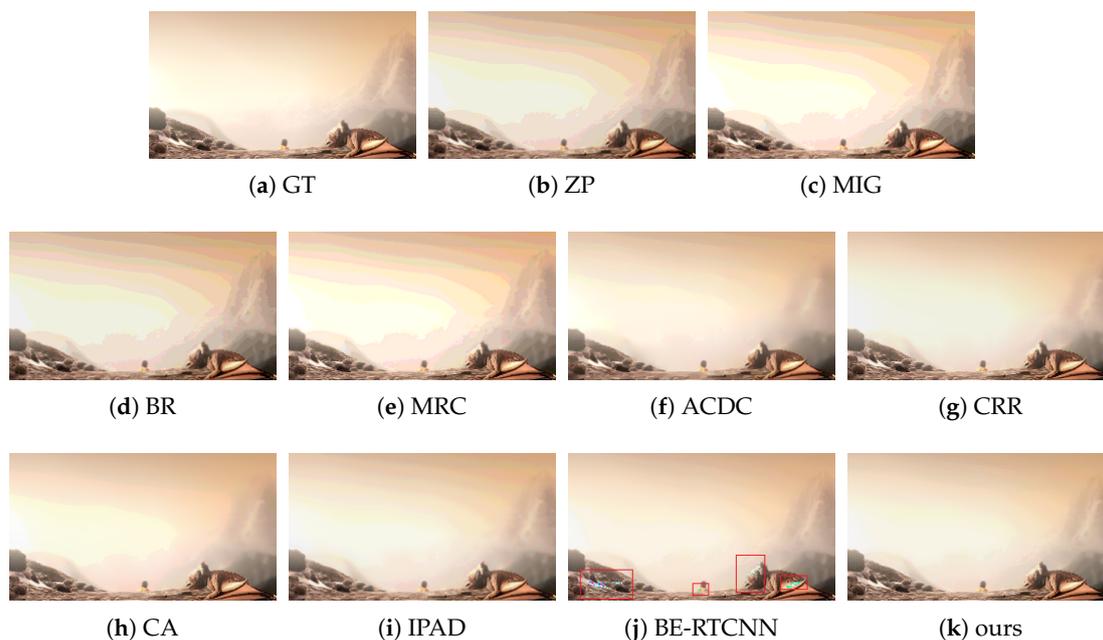


Figure 4. The BDE results on 4bit-16bit synthetic image from sintel-8 dataset [37].

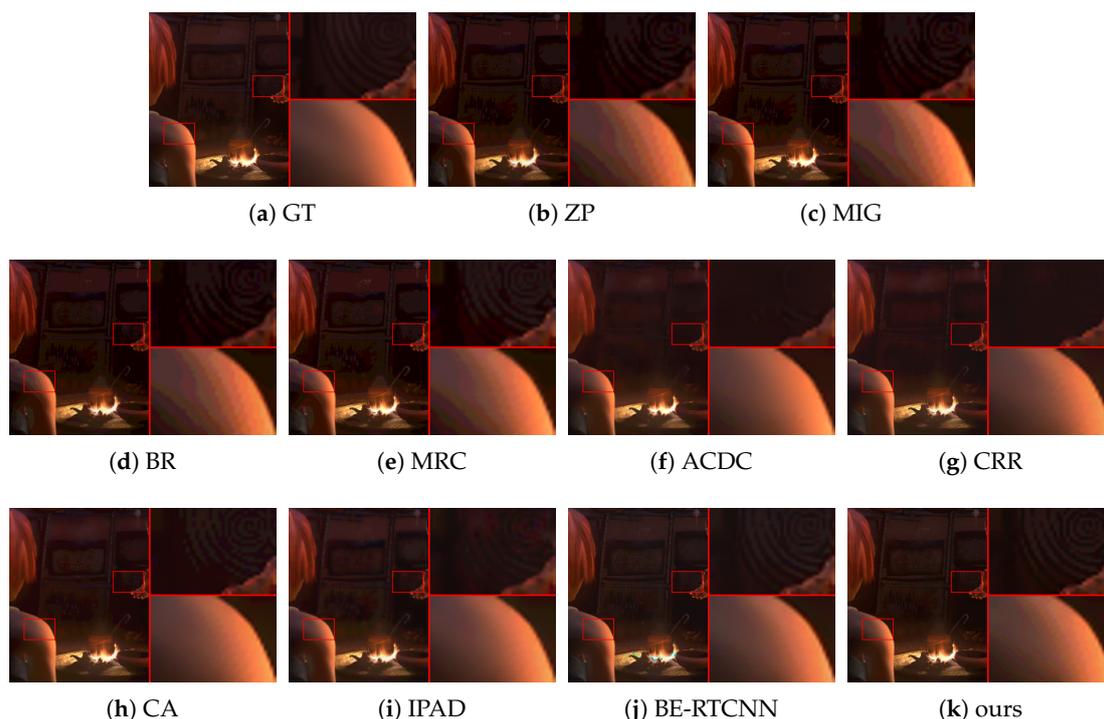


Figure 5. The BDE results on 4bit-16bit synthetic image from sintel-8 dataset.

In Figure 5, to better show the suppression effect of false contours and the degree of detail preservation by different methods, we enlarge one smooth area and detail area circled by the red boxes 4 times and paste them on the right side of the image. It can be seen from Figure 5b–e that ZP, MIG, BR and MRC perform poorly in suppressing false contours in smooth regions, but they can maintain the real contours of detail regions well. ACDC, CRR and IPAD are just the opposite. CA's detail maintenance is better than that of CRR, but it is still not satisfactory. BE-RTCNN does a good job of suppressing false contours and maintaining details, but it has obvious color distortion. Our method achieves the best tradeoff between false contour suppression, color distortion suppression, and detail preservation, thus producing the best visual quality.

Our algorithm has obvious advantages in false contour suppression and avoiding color distortion. The superiority of our method can also be seen from Figure 6. Compared with the original image, except for the method proposed in this paper, the bricks and zebra lines in the recovered image of all other algorithms have different degrees of tonal distortion. Color distortion occurred in the resulting image of BE-RTCNN. For better viewing, we randomly select one distortion region, zoom in it by 4 times, and stick it at the bottom right of the image with a red box to highlight it.

Figure 7a is one image fetched randomly from the Kodak data set. Figure 7b is the restored images by ZP. To further reveal the suppression effect of false contours, a row of data extracted from the smooth area of the recovered image and the original data of its corresponding position are plotted in Figure 7c–h for comparison. Obvious steps appear in the ZP's curve as shown by Figure 7c. Those steps indicate false contours. Steps are well smoothed in Figure 7d–f. This is consistent with the high PSNR listed in Table 1. BDE's curve (Figure 7g) has a similar shape with ground-truth but has obvious value offset from left to right. This may be because flat and non-flat areas are reconstructed separately, in order to make a smooth transition between flat and non-flat areas, and make the boundary of the regions more natural, so the fusion step made some kind of value convergence. As shown in Figure 7h, the result obtained by our method maintains good consistency with the ground-truth data in both the shape and numerical value.

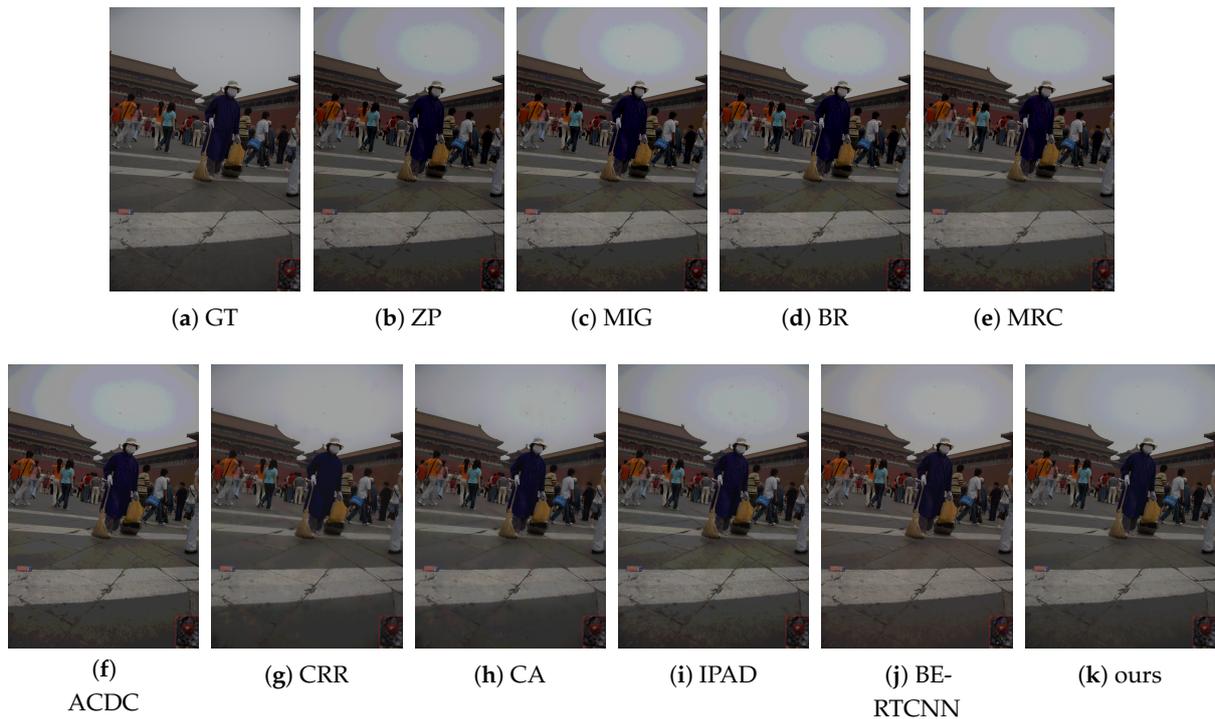


Figure 6. The BDE results on 4bit-16bit natural image from fiveK- dataset.

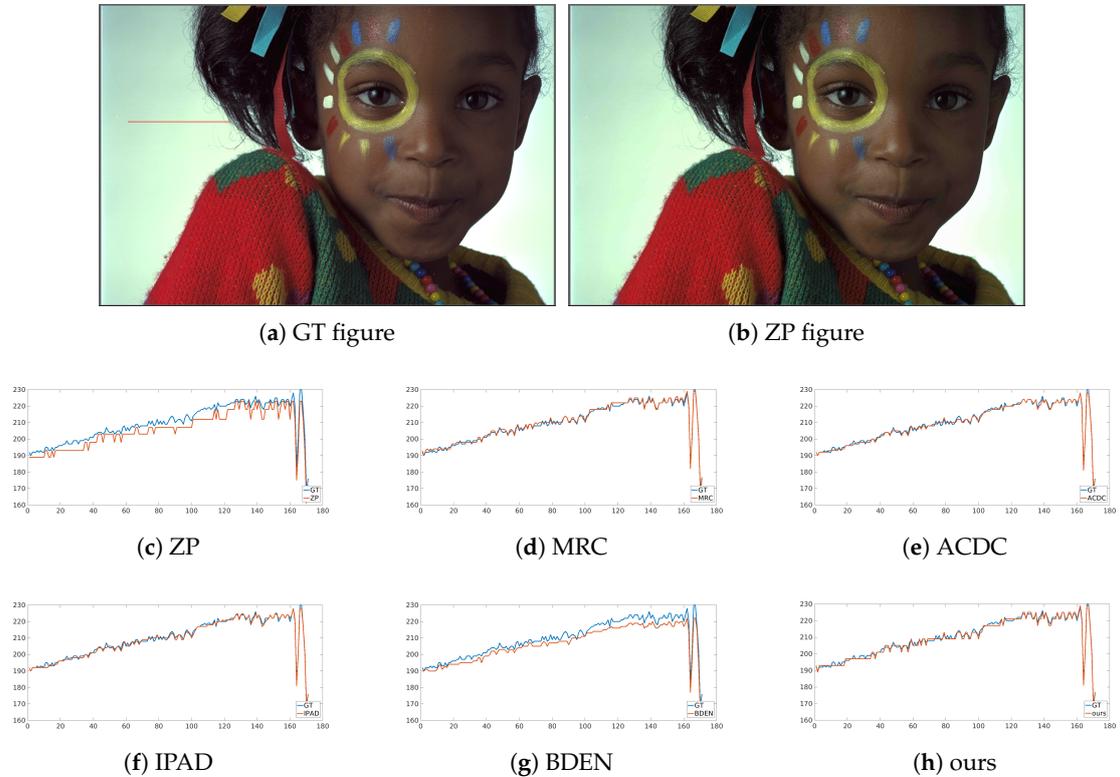


Figure 7. Intensity curves of 6bit-8bit natural BDE image from Kodak.

As shown in Table 1 and Figures 4–7, our method can suppress false contours and color distortion well, and preserve details at the same time, thus having state-of-the-art objective metric PSNRs and subjective visual perception quality. This comes from the

encoder-decoder network model implemented by convolution and deconvolution, global skip connection and the use of *vgg_loss*.

4.2.3. Run Time Comparison

To measure the computational complexity, we test the run times of the algorithms in comparison on the testset sintel-8 with resolution 500×218 . We test on a platform with Intel Core™ i5-7500 CPU @ 3.40 GHz \times 4 and Samsung DDR4 8 GB RAM @ 2666 MHz. Run times are listed in Table 2. As can be seen from Table 2 that our algorithm has a significant run time advantage over both traditional algorithms and related deep learning algorithms.

Table 2. Average run time (in seconds) of different algorithms.

Algorithm	ZP	MIG	BR [10]	MRC [11]	ACDC [28]	CRR [12]	CA [29]	IPAD [13]	BE-RTCNN [15]	BDEN [16]	Ours
Run time	0.0005	0.0012	0.0041	90.4498	332.0405	15.9587	24.476	7.2593	1.9025	2.6334	1.1031

4.3. Ablation Analysis

4.3.1. Analysis of Color Distortion

As we can see from these three sub-figures (j) of Figures 4–6, the results of BE-RTCNN [14] have obvious color distortion. This type of color distortion is not unique for BE-RTCNN, but shows some kind of universality for CNN-based BDE algorithms. To show this, we delete the global skip connection of BE-AUTO while keeping all the other components the same to get a new solution, which we call BE-AUTO-base. Figure 8c depicts the result of BE-AUTO-base. The same kind of color distortion as that of BE-RTCNN occurs as we compare it (Figure 8c) with the ground-truth (Figure 8a) and Figure 5j.

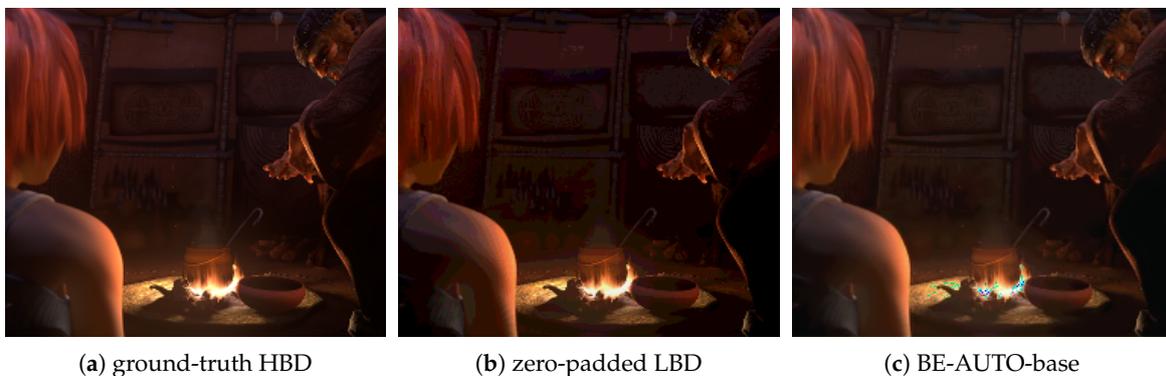


Figure 8. False contour suppression and color distortion [37]. (a) is the ground-truth 16-bit image; (b) is the zero-padded fake 16-bit image from 4-bit; and (c) is the result of BE-AUTO-base.

Visual quality would severely degrade if there is color distortion. BDE’s application scenarios, such as high-bit display, are hoped to present better visual quality, which means that the visual quality of the resulting image is an important metric for evaluating BDE algorithms. Therefore, to achieve better visual quality, both color distortion and false contours should be effectively suppressed.

In this section, we will analyze the causes of color distortion and the way to suppress color distortion without affecting the suppression of false contours. We will also show that the traditional truncation method cannot effectively solve this problem.

4.3.2. Color Distortion Reasons

Just as with other deep learning (such as EDSR [33]) algorithms for image processing, we preprocess the data range before training and testing to avoid interference with the

network by the numerical range of the input images. In this article, we will map the pixel values of the input and label images to $[-1,1]$ according to Equation (5). Therefore, it is reasonable to expect that the value range of the network's output is also $[-1,1]$. Moreover, because the data type is unsigned integer when the image is stored and displayed, the output of the network is processed by Equations (6) and (7) and then stored, displayed, and to compute objective metrics (such as PSNR).

$$\text{map}(X) = X/\text{max_value} - 1 \quad \text{where} \quad \text{max_value} = 2^p - 1 \quad (5)$$

$$\text{remap}(X) = (X + 1)/2 \times \text{max_value} \quad (6)$$

$$X = \text{uint}(X) \quad (7)$$

However, when we check the output of the BE-AUTO-base, we find that the value range of the output image exceeds the expected interval $[-1, 1]$. The numerical ranges of the eight output images are shown in Table 3. Once the value crosses the boundary and is processed by Equation (6), a value greater than max_value or smaller than 0 will appear. As is shown in Equation (8), after these out-of-bounds values are type-converted, numerical flip occurs, and the image's color will distort. In addition, if the estimated value of any color component significantly exceeds the interval determined by the true value, the ratio of the three color channels will be out of balance and results in color distortion.

$$\text{uint16}(x) = \begin{cases} \lfloor x \rfloor & x \in [0, 65,535] \\ \lfloor x - 65535 \rfloor & x \in (65,535, +\infty] \\ \lfloor x + 65536 \rfloor & x \in (-\infty, 0) \end{cases} \quad (8)$$

Table 3. Output range of BE-AUTO-base on 8 random selected images.

img_idx	img1	img2	img3	img4	img5	img6	img7	img8
range	$[-1.16,1.21]$	$[-0.94,0.76]$	$[-1.14,1.18]$	$[-1.18,1.01]$	$[-1.21,1.05]$	$[-1.17,1.04]$	$[-1.18,0.97]$	$[-1.29,1.18]$

4.3.3. Non-Triviality of Color Distortion Problem

Convolution neural network is a process of sliding convolution. To explore the underlying reasons for the values exceeding the range, we take a convolution operation as an example. Also, for ease of analysis, the bias and activation of the convolution neural network are ignored. As is shown in Figure 9, the left part is the image block to be convolved and the right part is a 3×3 convolution kernel. Therefore, the convolution operation can be expressed by Equation (9):

$$y = \sum_{i=0}^8 \omega_i \times x_i \quad (9)$$

where y is the output value, x_i is the input pixel, w_i is the weight, and the index number i is as shown in Figure 9. The ground-truth value of y called \hat{y} lies in $[x_4, x_4 + b_g - 1]$, where $b_g = 2^g$ is the quantization step, $g = p - q$ is the difference between the target bit-depth p and the source bit-depth q . Therefore, the training process of the convolution neural network essentially seeks a set of parameters W to map each group of X to a variation interval $[x_4, x_4 + b_g - 1]$. Since x_i belongs to $[0, b_g * \text{max}_q]$, where $\text{max}_q = 2^q - 1$ is the maximum pixel value of the source image, and the target range T for a given pixel x_4 is

$$T = [x_4, x_4 + b_g - 1] \quad (10)$$

then the target range for the whole image UT is the union of all T , i.e.,

$$UT = \cup_{x \in [0, b_g \times \max_q]} T = [0, b_g \times \max_q + b_g - 1] = [0, 2^p - 1] \quad (11)$$

Taking 4-bit to 16-bit BDE as an example, $UT = [0, 2^{16} - 1] = [0, 65, 535]$, which means that the neural network needs to have the ability to map the input image to $[0, 65, 535]$, although the value of each output pixel is theoretically only within $T = [x_4, x_4 + b_{12} - 1] = [x_4, x_4 + 4095]$. T and UT have a huge difference in the length of the interval. Moreover, the convolution neural network not only needs to cope with the prediction of a certain pixel but with different images and different pixels. Therefore, to minimize the average prediction error over the entire image and throughout the data set for better generalization ability, the training process must focus on UT and ignore T . So, when T and UT are different, it is bound to sacrifice prediction accuracy for a certain pixel. This kind of sacrifice is reflected as that the prediction result is not within the interval $[x_4, x_4 + b_g - 1]$ in the prediction of a certain pixel, and is not in the interval $[0, 2^p - 1]$ when predicting for multiple pixels or the whole picture. This out-of-bounds eventually manifests itself as color distortion. Therefore, to suppress color distortion, the difference between T and UT should be minimized and be identical in the ideal situation. From Equations (10) and (11), we can see that when T is independent of x_4 , $UT = T$.

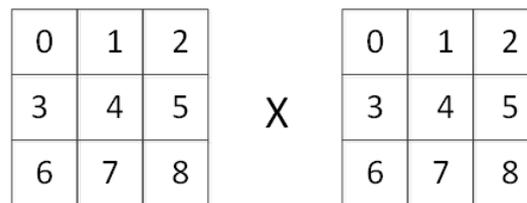


Figure 9. Example of image patch and convolution kernel. The number in each box is the index of corresponding pixel or the convolution kernel's weight.

The color distortion problem can also be analyzed from the perspective of estimation error. We decompose a p -bit prediction of the value y into the most significant q -bits and residual g -bits, called h , r . therefore:

$$y = b_g \times h + r \quad (12)$$

Among them, h lies in $[0, 2^q - 1]$, and r lies in $[0, 2^g - 1]$. Performing error analysis on y , we can get:

$$\Delta y = b_g \times \Delta h + \Delta r \quad (13)$$

When $p = 16, q = 4$, Equation (13) is instantiated as $\Delta y = 4096 \times \Delta h + \Delta r$. From Equation (13), we can see that a very small error in h can make a large error in the estimation of y because Δh is significantly magnified by b_g . Therefore, to effectively suppress color distortion, it is necessary to suppress the estimation error of the most significant 4 bits. If an accurate estimate of the 4 bits is obtained, the numerical out-of-bounds and color distortion can be effectively suppressed. We can see that the correct estimate of the high 4 bits h is equal to the high 4 bits of x_4 , because x_4 is obtained by bit-shifting the 16 bits \hat{y} by 12 bits, the high 4 bits of x_4 and \hat{y} are identical). Therefore, when estimating y , if we can get x_4 , we only need to estimate the lower 12-bits, and we can make T independent of x_4 .

In previous CNN-based algorithms, the network output is truncated. That is, the value not within the interval is forcibly assigned to the nearest interval boundary value. This operation is based on the prior that the value represented by finite bits is limited. Taking p -bit image as an example, the range of pixel values that can be represented is $[0, 2^p - 1]$, so predicted values greater than $2^p - 1$ or less than 0 are false predictions. Re-assigning these out-of-bounds values to the nearest interval boundary not only enables the predicted value to be represented by the finite bits but also reduces the estimation

error. However, with the analysis of the out-of-bounds causes in the above two paragraphs, we can find that the numerical truncation can only deal with the out-of-bounds for the full interval $[0, 2^p - 1]$, but cannot deal with the out-of-bounds for $[x_4, x_4 + b_g - 1]$ that occurs when predicting certain pixel. Therefore, numerical truncation can only be used as an auxiliary solution. To better resolve numerical out-of-bounds and color distortion, we should minimize the estimated error of q -bits or make T unrelated to x_4 , and in this paper, we propose to use global skip to achieve this.

4.3.4. Effect of Global Skip Connections

The difference between BE-AUTO and BE-AUTO-base is that the former has a global skip connection. This is a very simple operation, but it is very effective. Moreover, the global skip connection is universal for the suppression of color distortion and can stably improve network performance. Global skip connections have also been used in other network structures, such as VDSR [32] in the image super-resolution domain, and the recent deep BDE algorithm BDEN [16], but neither functionality nor universality of global skip connections are proven. We will compare and verify on the existing three deep BDE networks (BE-AUTO, BE-RTCNN, BDEN). Among them, BE-RTCNN has no global skip connection, and BDEN has. Because BE-RTCNN does not provide code, and BDEN's code is an encrypted file, we implemented it as described by the papers. In Figure 10, BE-RTCNN* and BDEN* refer to our implemented version, and BE-RTCNN-AG refers to the addition of a global skip connection. BDEN-DG refers to the deletion of the global skip connection. Figure 10 is the *Cumulative Distribution Function* (CDF) of the number of out-of-bounds pixels for three network structures with and without global skip connection on 1909 randomly selected images of size 436×1024 . The average number of out-of-bounds pixels are listed in Table 4. It can be seen from Figure 10 that for each network structure, the output value range of scheme without the global skip connection has a significant out-of-bounds, and after the global skip connection is added, it can be well maintained within the predetermined interval, avoiding color distortion. It can be seen that the global skip connection has a significant effect on the definition of the output value range and the avoidance of color distortion. This effect is independent of the network structure and is universal. Therefore, when performing a deep BDE network design, the global skip connection can be used as a regular component. The experimental results here are consistent with the analysis in Section 4.3.1.

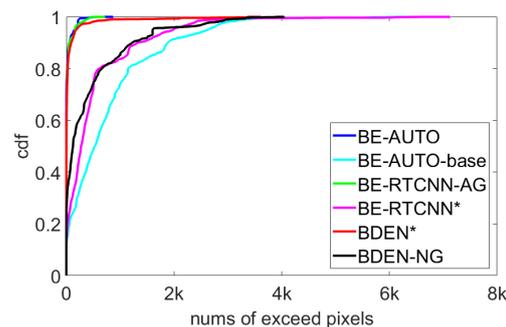


Figure 10. CDF of the number of out-of-bounds pixels for three network structures with and without global skip connections. * refers to our implemented version.

Table 4. Average numbers of out-of-bounds pixels of three network structures on 1909 randomly selected images with and without global skip connections.

	BDEN	BE-RTCNN	Ours
without global skip	415.62	495.34	761.04
with global skip	53.13	26.36	20.27

4.3.5. Convolution vs. Deconvolution

Convolution and deconvolution are a set of reverse operations. Convolution can be seen as a “bottom-up” process, such as extracting edges, contours and even objects from low-level features to high-level features from the input image. Deconvolution is a “top-down” [40] process. The idea of deconvolution is to backtrack what kind of low-level features can produce current characteristics. The level-by-level abstraction of convolution and step-by-step reconstruction of deconvolution corresponds to the encoding module and decoding module of the auto-encoder well, which is the basic idea of designing BE-AUTO. This idea has also been verified experimentally. In Figure 11, full-conv means that the codec module is implemented with convolution layers. Full-deconv means that the codec is implemented by deconvolution. In addition, conv-deconv refers to encode with convolution and decode with deconvolution layers, which is the scheme adopted by BE-AUTO. These three schemes are consistent except for the convolution or deconvolution layers. Figure 11 shows the CDF of PSNR of these three implementations on 1886 images selected randomly from sintel-2000 and fiveK. The average PSNRs of these three scenarios are listed in Table 5. As we can see from Figure 11, conv-deconv scheme has a significant advantage over the full-conv, because conv-deconv’s CDF curve (the red line) lies at the right side of the full-conv’s (the pink line) with a big gap. The full-deconv scheme (the blue line) has similar performance to that of the conv-deconv scheme at the low PSNR level, but gradually falls behind with the increase of PSNR. This shows that the conv-deconv solution can stably improve the reconstruction quality of the test image compared with the other two schemes, so we chose conv-deconv (i.e., convolution-combined-with-deconvolution) at the final scheme.

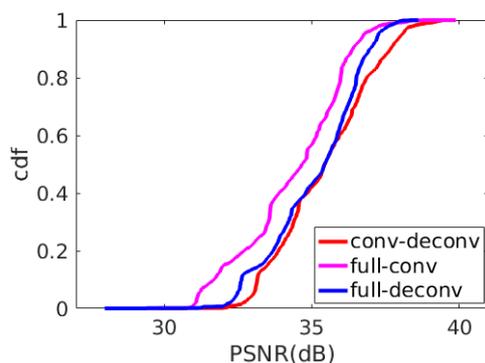


Figure 11. CDF of PSNR for network structure implemented with convolution or deconvolution on 1886 random selected images.

Table 5. PSNR of conv / deconv on 1886 random selected images.

	<i>Full_Conv</i>	<i>Full_Deconv</i>	<i>Conv_Deconv</i>
average PSNR	34.3741	35.1176	35.4293

5. Conclusions

In this paper, an auto-encoder-like CNN model is proposed to suppress color distortion and false contours in images of bit-depth enhancement. Based on this model, a simple bit-depth re-quantization method can be used in PFEs of visual IoT to significantly decrease the visual data volumes with extremely low computation complexity and low energy consumption and achieve high visual perception performance of the restored visual image. The obtained experimental results reveal that our method can obtain better results compared to other competing methods in subjective and objective performance. Future work will focus on integrating the model into HEVC reference frame and reducing the calculation complexity of the model.

Author Contributions: C.P. main contribution was to propose the methodology of this work and write the paper. L.C. participated in the implementation of the algorithm. X.H. was responsible for algorithm validation and data analysis. Z.F., J.X. and X.L. guided the entire research process and secured funding for the research. All authors have read and agreed to the published version of the manuscript.

Funding: This document is the results of the research project funded by the National Natural Science Foundation of China (61671126).

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: <http://r0k.us/graphics/kodak/>, <https://www.xiph.org/>, and at <https://data.csail.mit.edu/graphics/fivek/>.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mehmood, I.; Ullah, A.; Muhammad, K.; Deng, D.J.; Meng, W.; Al-Turjman, F.; Sajjad, M.; de Albuquerque, V.H.C. Efficient Image Recognition and Retrieval on IoT-Assisted Energy-Constrained Platforms From Big Data Repositories. *IEEE Internet Things J.* **2019**, *6*, 9246–9255. [[CrossRef](#)]
2. Ji, W.; Xu, J.; Qiao, H.; Zhou, M.; Liang, B. Visual IoT: Enabling internet of things visualization in smart cities. *IEEE Netw.* **2019**, *33*, 102–110. [[CrossRef](#)]
3. Internet of Business. Opinion: The Visual Internet of Things—Why IoT Needs Visual Data. 2013. Available online: <https://internetofbusiness.com/opinion-the-visual-internet-of-things-why-iot-needs-visual-data/> (accessed on 1 May 2019).
4. Zhang, Z.; Li, D. Hybrid Cross Deep Network for Domain Adaptation and Energy Saving in Visual Internet of Things. *IEEE Internet Things J.* **2019**, *6*, 6026–6033. [[CrossRef](#)]
5. Zhang, Y.; He, Q.; Xiang, Y.; Zhang, L.Y.; Liu, B.; Chen, J.; Xie, Y. Low-cost and confidentiality-preserving data acquisition for internet of multimedia things. *IEEE Internet Things J.* **2017**, *5*, 3442–3451. [[CrossRef](#)]
6. Fragkiadakis, A.; Charalampidis, P.; Tragos, E. Adaptive compressive sensing for energy efficient smart objects in IoT applications. In Proceedings of the 2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), Aalborg, Denmark, 11–14 May 2014; pp. 1–5.
7. Li, S.; Da Xu, L.; Wang, X. Compressed sensing signal and data acquisition in wireless sensor networks and internet of things. *IEEE Trans. Ind. Inform.* **2012**, *9*, 2177–2186. [[CrossRef](#)]
8. Dixon, A.M.; Allstot, E.G.; Gangopadhyay, D.; Allstot, D.J. Compressed sensing system considerations for ECG and EMG wireless biosensors. *IEEE Trans. Biomed. Circuits Syst.* **2012**, *6*, 156–166.
9. Peng, H.; Yang, B.; Li, L.; Yang, Y. Secure and Traceable Image Transmission Scheme Based on Semitensor Product Compressed Sensing in Telemedicine System. *IEEE Internet Things J.* **2020**, *7*, 2432–2451.
10. Ulichney, R.; Cheung, S. Pixel Bit-Depth Increase by Bit Replication. *Proc. SPIE Int. Soc. Opt. Eng.* **1998**, *3300*, 232–241. [[CrossRef](#)]
11. Mittal, G.; Jakhetiya, V.; Jaiswal, S.; Au, O.; Tiwari, A.; Dei, W. Bit-depth expansion using Minimum Risk Based Classification. In Proceedings of the 2012 Visual Communications and Image Processing, San Diego, CA, USA, 27–30 November 2012; pp. 1–5. [[CrossRef](#)]
12. Hong Cheng, C.; Au, O.; Liu, C.H.; Yue Yip, K. Bit-depth Expansion by Contour Region Reconstruction. In Proceedings of the 2009 IEEE International Symposium on Circuits and Systems, Taipei, Taiwan, 24–27 May 2009; pp. 944–947. [[CrossRef](#)]
13. Liu, J.; Zhai, G.; Liu, A.; Yang, X.; Zhao, X.; Chen, C.W. IPAD: Intensity Potential for Adaptive De-Quantization. *IEEE Trans. Image Process.* **2018**, *27*, 4860–4872. [[CrossRef](#)]
14. Liu, J.; Sun, W.; Liu, Y. Bit-Depth Enhancement via Convolutional Neural Network. In *Digital TV and Wireless Multimedia Communication*; Zhai, G., Zhou, J., Yang, X., Eds.; Springer: Singapore, 2018; pp. 255–264.
15. Su, Y.; Sun, W.; Liu, J.; Zhai, G.; Jing, P. Photo-realistic image bit-depth enhancement via residual transposed convolutional neural network. *Neurocomputing* **2019**, *347*, 200–211.
16. Zhao, Y.; Wang, R.; Jia, W.; Zuo, W.; Liu, X.; Gao, W. Deep Reconstruction of Least Significant Bits for Bit-Depth Expansion. *IEEE Trans. Image Process.* **2019**, *28*, 2847–2859. [[CrossRef](#)] [[PubMed](#)]
17. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.
18. Endo, Y.; Kanamori, Y.; Mitani, J. Deep Reverse Tone Mapping. *ACM Trans. Graph.* **2017**, *36*, 177.
19. Liu, J.; Sun, W.; Jing, P.; Yu, J.; Su, Y. Improving Bit-depth Expansion via Context-aware MMSE Optimization (CAMO). *IEEE Access* **2018**, *6*, 46396–46406. [[CrossRef](#)]
20. Woo, J.; Lee, S.; Choe, W. Bit Depth Expansion via Estimation of Bit Value Expectation. *Electron. Imaging* **2016**, *2016*, 1–6. [[CrossRef](#)]

21. Wang, Y.; Huang, H.; Wang, C.; He, T.; Wang, J.; Hoai, M. Gif2video: Color dequantization and temporal interpolation of gif images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–21 June 2019; pp. 1419–1428.
22. Gupta, P.S.; Yuan, X.; Choi, G.S. LISR: Image Super-resolution under Hardware Constraints. *arXiv* **2019**, arXiv:1909.10136.
23. Taguchi, A.; Nishiyama, J. Bit-length expansion by inverse quantization process. In Proceedings of the 2012 20th European Signal Processing Conference (EUSIPCO), Bucharest, Romania, 27–31 August 2012; pp. 1543–1547.
24. Liu, J.; Liu, P.; Su, Y.; Jing, P.; Yang, X. Spatiotemporal Symmetric Convolutional Neural Network for Video Bit-Depth Enhancement. *IEEE Trans. Multimed.* **2019**, *21*, 2397–2406. [[CrossRef](#)]
25. Wan, P.; Au, O.; Tang, K.; Guo, Y. Image de-quantization via spatially varying sparsity prior. In Proceedings of the 2012 19th IEEE International Conference on Image Processing, Orlando, FL, USA, 30 September–3 October 2012; pp. 953–956. [[CrossRef](#)]
26. Umeda, S.; Watanabe, H.; Ikai, T.; Hashimoto, T.; Chujoh, T.; Ito, N. Joint super-resolution and bit depth extension by DNN. In *International Workshop on Advanced Image Technology (IWAIT)*; International Society for Optics and Photonics: Bellingham, WA, USA, 2019; Volume 11049, p. 1104925.
27. Wan, P.; Cheung, G.; Florencio, D.; Zhang, C.; Au, O. Image bit-depth enhancement via maximum-a-posteriori estimation of graph AC component. In Proceedings of the 2014 IEEE International Conference on Image Processing ICIP, Paris, France, 27–30 October 2014; Volume 25, pp. 4052–4056. [[CrossRef](#)]
28. Wan, P.; Cheung, G.; Florencio, D.; Zhang, C.; Au, O.C. Image Bit-Depth Enhancement via Maximum A Posteriori Estimation of AC Signal. *IEEE Trans. Image Process.* **2016**, *25*, 2896–2909. [[CrossRef](#)]
29. Wan, P.; Au, O.; Tang, K.; Guo, Y.; Fang, L. From 2D Extrapolation to 1D Interpolation: Content Adaptive Image Bit-Depth Expansion. In Proceedings of the 2012 IEEE International Conference on Multimedia and Expo, Melbourne, VIC, Australia, 9–13 July 2012; pp. 170–175. [[CrossRef](#)]
30. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 295–307. [[CrossRef](#)]
31. Dong, C.; Loy, C.C.; Tang, X. Accelerating the super-resolution convolutional neural network. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 391–407.
32. Kim, J.; Kwon Lee, J.; Mu Lee, K. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1646–1654.
33. Lim, B.; Son, S.; Kim, H.; Nah, S.; Mu Lee, K. Enhanced deep residual networks for single image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 136–144.
34. Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A.P.; Bishop, R.; Rueckert, D.; Wang, Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1874–1883.
35. Liu, J.; Sun, W.; Su, Y.; Jing, P.; Yang, X. BE-CALF: Bit-depth enhancement by concatenating all level features of DNN. *IEEE Trans. Image Process.* **2019**, *28*, 4926–4940. [[CrossRef](#)]
36. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
37. Peng, C.; Cai, L.; Fu, Z.; Li, X. CNN-based bit-depth enhancement by the suppression of false contour and color distortion. In Proceedings of the 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Lanzhou, China, 18–21 November 2019; pp. 1145–1151.
38. Ulyanov, D.; Vedaldi, A.; Lempitsky, V. Deep image prior. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9446–9454.
39. Zhao, H.; Gallo, O.; Frosio, I.; Kautz, J. Loss Functions for Neural Networks for Image Processing. *arXiv* **2015**, arXiv:1511.08861.
40. Zeiler, M.D.; Krishnan, D.; Taylor, G.W.; Fergus, R. Deconvolutional networks. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2528–2535. [[CrossRef](#)]
41. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32Nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; Volume 37, pp. 448–456.
42. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
43. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the fourteenth International Conference on Artificial Intelligence and Statistics, Ft. Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
44. Kodak. Kodak Lossless True Color Image Suite. 2013. Available online: <http://r0k.us/graphics/kodak/> (accessed on 1 May 2019).
45. X.Foundation. Xiph.Org. 2016. Available online: <https://www.xiph.org/> (accessed on 1 May 2019).
46. Bychkovsky, V.; Paris, S.; Chan, E.; Durand, F. Learning photographic global tonal adjustment with a database of input/output image pairs. In Proceedings of the CVPR 2011, Providence, RI, USA, 20–25 June 2011; pp. 97–104.
47. Agustsson, E.; Timofte, R. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Salt Lake City, UT, USA, 28 July 2017; pp. 1122–1131.